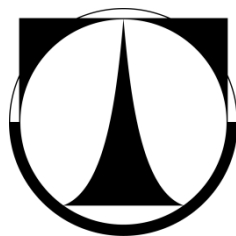


**TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií



**BAKALÁŘSKÁ PRÁCE**

Liberec 2012

**David Tauchmann**

# **TECHNICKÁ UNIVERZITA V LIBERCI**

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2646 – Informační technologie

Studijní obor: 1802R007 – Informační technologie

## **Monitoring využívání síťových licencí**

### **Network License Monitor**

#### **Bakalářská práce**

Autor:	David Tauchmann
Vedoucí práce:	Ing. Jiří Jeníček, Ph.D.

V Liberci 14. 5. 2012

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **David Tauchmann**  
Osobní číslo: **M09000186**  
Studijní program: **B2646 Informační technologie**  
Studijní obor: **Informační technologie**  
Název tématu: **Monitoring využívání síťových licencí**  
Zadávající katedra: **Ústav informačních technologií a elektroniky**

### Z á s a d y   p r o   v y p r a c o v á n í :

1. Seznamte se s fungováním licenčního serveru a s možnostmi jeho monitorování.
2. Pomocí Open-source technologií navrhňte a realizujte webovou aplikaci pro monitorování využití licenčních serverů a jednotlivých licencí.
3. Aplikaci propojte s licenčním serverem existujícím na TUL.

Rozsah grafických prací: Dle potřeby dokumentace  
Rozsah pracovní zprávy: cca 30 stran  
Forma zpracování bakalářské práce: tištěná/elektronická  
Seznam odborné literatury:

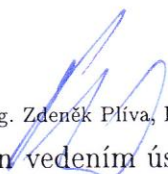
- [1] Gilmore, W. J.: Velká kniha PHP a MySQL 5, Zoner Press, ISBN 80-86815-53-6
- [2] Boronczyk, T. et al.: PHP 6, MySQL, Apache - Vytváříme webové aplikace, ISBN 978-80-251-2767-4
- [3] FlexNet Publisher, online: <http://www.flexerasoftware.com>

Vedoucí bakalářské práce: Ing. Jiří Jeníček, Ph.D.  
Ústav informačních technologií a elektroniky

Datum zadání bakalářské práce: 1. října 2011  
Termín odevzdání bakalářské práce: 18. května 2012

  
prof. Ing. Václav Kopecký, CSc.  
děkan



  
prof. Ing. Zdeněk Plíva, Ph.D.  
pověřen vedením ústavu

V Liberci dne 1. října 2011

## **Prohlášení**

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé diplomové práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum

Podpis

## **Poděkování**

Rád bych upřímně poděkoval svému vedoucímu *Ing. Jiřímu Jeníčkovi, Ph.D.* za vedení této bakalářské práce, cenné rady, doporučení a věcné připomínky.

## **Abstrakt**

Bakalářská práce je zaměřena na problematiku využívání síťových licencí v univerzitním prostředí. Obsahuje charakteristiku licenčního serveru FLEXlm. Odpovědí na potřebu sledovat historii využití licenčního serveru je realizace webové aplikace, která zajišťuje průběžné shromažďování, analýzu a vizualizaci dat získaných z licenčního serveru. Nedílnou součástí této práce je rozbor navržené aplikace včetně použitých technologií a popis implementace na licenční server.

## **Klíčová slova**

licenční server, plovoucí síťová licence, FLEXlm

## **Abstract**

The bachelor thesis specialises in matters of using network licenses on university background. It contains characteristics of license server FLEXlm. The answer for the need of observing the history of using the license server is implementation of web application, which provides continuous license data collecting, analysis and visualisation. The analysis of suggested application including applied technologies and description of license server implementation is an inherent part of this thesis.

## **Keywords**

license server, floating network license, FLEXlm

# Obsah

Prohlášení.....	3
Poděkování.....	4
Abstrakt.....	5
Abstract.....	5
Obsah .....	6
Seznam obrázků.....	9
Seznam tabulek .....	9
Seznam zkratk a termínů.....	10
1 Úvod a motivace .....	12
2 Licenční server FLEXlm .....	14
2.1 Charakteristika .....	14
2.2 Úkoly licenčního serveru .....	14
2.3 Princip činnosti.....	14
2.4 Komponenty licenčního serveru .....	16
2.4.1 Démon správce licencí ( <i>lmgrd</i> nebo <i>lmgrd.exe</i> ).....	16
2.4.2 Démon dodavatele softwaru ( <i>vendor daemon</i> ).....	16
2.4.3 Licenční soubor ( <i>licence.dat</i> ) .....	16
2.4.4 Licencovaná aplikace.....	16
2.5 Způsoby implementace .....	16
2.5.1 Model s jedním licenčním serverem .....	16
2.5.2 Model s distribuovanými licenčními servery.....	17
2.5.3 Model s redundantními licenčními servery .....	17
3 Analýza existujících řešení .....	18
3.1 Komerční aplikace .....	18
3.1.1 FlexNet Manager .....	18
3.1.2 Alternativní komerční řešení .....	19
3.2 Open-source aplikace .....	19
3.2.1 phpLicenseWatcher .....	20
3.2.2 flexmonitor.....	20
4 Webová aplikace.....	21
4.1 Použité technologie .....	21



4.1.1	PHP .....	21
4.1.2	JpGraph.....	21
4.1.3	MySQL .....	21
4.1.4	jQuery .....	22
4.1.5	XHTML .....	23
4.1.6	Vzhled aplikace.....	23
4.2	Dokumentace v kódu.....	23
4.3	Architektonický vzor.....	24
4.4	Aplikační logika .....	24
4.4.1	controller_base.class.php .....	24
4.4.2	registry.class.php .....	24
4.4.3	router.class.php .....	25
4.4.4	template.class.php .....	25
4.4.5	user.class.php.....	25
4.5	Funkce aplikace.....	26
4.5.1	Úvodní přehledová strana .....	26
4.5.2	Aktuální využití .....	27
4.5.3	Historie využití .....	28
4.5.4	Platnost features .....	29
4.5.5	Administrace .....	30
4.5.6	Přihlašování, odhlašování a změna hesla uživatele .....	32
5	Implementace na licenční server a měření výkonu.....	33
5.1	Požadavky .....	33
5.1.1	Apache .....	33
5.1.2	PHP .....	33
5.1.3	JpGraph.....	34
5.1.4	MySQL .....	34
5.2	Konfigurace aplikace .....	35
5.3	Dokončení instalace .....	35
5.4	Analýza požadavků na velikost diskového prostoru.....	36
5.5	Testování aplikace v průběhu vývoje.....	37
5.6	Měření výkonu na licenčním serveru .....	39
	Závěr .....	41

Seznam použité literatury .....	42
Seznam příloh .....	44
Příloha A – Metoda validující e-mailovou adresu .....	45
Příloha B – Šifrovací funkce.....	46
Příloha C – Úvodní přehledová strana .....	47
Příloha D – Statistika aktuálního využití .....	48
Příloha E – Zobrazení historie využití konkrétní licence .....	49
Příloha F – Platnost zakoupených licencí .....	50
Příloha G – Administrační rozhraní.....	51
Příloha H – ER diagram databáze .....	52

## Seznam obrázků

Obr. 2.1: Průběh žádosti o vydání plovoucí síťové licence [4] .....	15
Obr. 4.1: Principiální schéma architektonického vzoru Model-View-Controller .....	24
Obr. C: Snímek úvodní strany webové aplikace.....	47
Obr. D: Snímek informující o využití licenčního serveru .....	48
Obr. E: Snímek obrazovky s grafem historie využití.....	49
Obr. F: Snímek zachycující platnost licencí .....	50
Obr. G: Snímek popisující rozložení administračního rozhraní .....	51
Obr. H: Databázový diagram popisující vztahy mezi entitami.....	52

## Seznam tabulek

Tabulka 1: Doba zpracování jednotlivých požadavků naměřená programem ab .....	38
Tabulka 2: Výsledky měření rychlosti webové aplikace na licenčním serveru.....	40

## Seznam zkratek a termínů

**cache (webová)** = vyrovnávací paměť určená pro dočasné uložení dat za účelem snížení zátěže a zvýšení rychlosti odezvy.

**cookie** = mechanismus, který řeší problém bezstavovosti protokolu HTTP.

V principu jsou informace, v něm uložené, přenášeny mezi serverem a klientem prostřednictvím malého textového souboru.

**démon** = služba, která běží na pozadí operačního systému, obsluhuje příchozí požadavky nebo je předává dalším procesům či programům k vyřízení. [2]

**démon dodavatele** = služba, která monitoruje vydané licence dodavatele a počítače, které tyto licence používají. [2]

**distribuovaný server** = model konfigurace licenčního serveru, který pro správu plovoucích síťových licencí používá celkem tři licenční servery. Každý distribuovaný server má unikátní licenční soubor a pevný počet licencí. [2]

**feature** = oddělená vlastnost, přidaná funkcionalita nebo komponenta softwaru.

**FLEXlm** = označení pro manažera licenčního serveru společnosti Flexera Software.

**FLEXnet** = technologie správy licencí společnosti Flexera Software. [2]

**framework** = „softwarová struktura, která slouží jako podpora při programování, vývoji a organizaci jiných softwarových projektů. Obvykle obsahuje podpůrné programy, knihovnu API a návrhové vzory.“ [11]

**heartbeat** = signál mezi licenčním serverem a licencovanou aplikací.

**JavaScript** = multiplatformní objektově orientovaný skriptovací jazyk, který se interpretuje na straně klienta. [7]

**licenční soubor** = soubor používaný licenčním serverem FLEXlm specifikující počet zakoupených licencí. [2]

**licenční server** = server zajišťující správu a distribuci plovoucích síťových licencí. [2]

**log soubor** = soubor, který obsahuje informace o výsledcích operací. Záznamy, v něm uložené, pomáhají administrátorovi analyzovat příčinu problému.

**MySQL** = multiplatformní relační databázový systém. Komunikace probíhá prostřednictvím dotazovacího jazyka SQL.

**PHP** = rekurzivní zkratka HyperText Preprocessor označuje skriptovací jazyk určený zejména pro programování dynamických webových stránek. [7]

**plovoucí síťová licence** = licence poskytovaná licenčním serverem, která není vázána na konkrétní pracovní stanici.

**pracovní stanice** = zde osobní počítač připojený do počítačové sítě.

**redundantní server** = model konfigurace licenčního serveru, který pro správu plovoucích síťových licencí používá celkem tři licenční servery. Tyto servery navzájem sdílejí licenční soubor i licenční fond. [2]

**RRDtool** = open-source standard pro ukládání a vykreslování velkého množství časově závislých dat. [10]

**session** = mechanismus, který řeší problém bezstavovosti protokolu HTTP. Zahrnuje množinu proměnných, které se předávají po celou dobu trvání relace. Po skončení skriptu jsou session proměnné uloženy na serveru.

**shelfware** = označení pro málo využívaný nebo nepoužívaný software.

**správce síťových licencí (lmgrd)** = démon, který zprostředkovává spojení mezi licencovanou aplikací a démonem dodavatele.

## 1 Úvod a motivace

Ve světě svobodného softwaru stanovuje licence pravidla, za kterých smí být s programem nakládáno. U komerčních počítačových programů slouží licence jako přístupový klíč, který odemyká plnou funkčnost programu a zároveň opravňuje uživatele k jeho používání. Dále bude licence chápána jako oprávnění k používání příslušného počítačového programu.

Tradiční způsob licencování předpokládá přítomnost platné licence na pevném disku každé pracovní stanice, kde je aplikace nainstalována. Tento přístup nachází uplatnění zejména v případech, kdy aplikace slouží omezenému počtu uživatelů na malém množství pracovních stanic. Komerční programy, které se spoléhají na tento způsob licencování, jsou tak určeny zejména pro používání v domácnostech a malých společnostech, kde jsou náklady na jejich administraci minimální. Jestliže si fyzická nebo právnická osoba zakoupí určité množství licencí k příslušnému počítačovému programu, má právo používat tento program pouze na těch pracovních stanicích, kde jsou tyto licence uloženy. V případě, že se program na jedné pracovní stanici aktuálně nepoužívá, nemůže takto uvolněnou licenci využít další uživatel na jiné pracovní stanici.

Typickými zástupci tradičního způsobu licencování jsou komerční programy dostupné pod doživotní licencí, u kterých se nepředpokládá častá migrace mezi pracovními stanicemi v doméně. Podobný způsob licencování používá například firma Microsoft v rámci svého operačního systému Windows ve verzích určených pro domácnosti a malé firmy. V jejich podání se jedná o nepřenositelný typ licence označovaný jako OEM. OEM licence je vázána na konkrétní počítač, potažmo na jeho základní desku.

Výše zmiňované omezení odstraňuje následující způsob licencování softwaru. V případě vysokého počtu pracovních stanic rostou náklady nejen na pořízení velkého počtu licencí, ale zejména na správu těchto licencí. Jeden ze způsobů, jak minimalizovat pořizovací náklady, nabízejí přímo někteří výrobci softwaru formou tzv. zvýhodněných multilicencí. Daleko efektivnějším řešením je pořízení licenčního serveru, který umožňuje centralizovanou správu licencí pro různé druhy softwaru. Mimo jiné dále zajišťuje distribuci plovoucích síťových licencí, která spočívá v tom, že licence jsou dynamicky přidělovány pracovním stanicím až v okamžiku, kdy o to požádají. Důsledkem tohoto přístupu je možnost optimalizovat počet licencí, eliminovat

*shelfware* a minimalizovat tak náklady s tím spojené. Mezi další klíčové argumenty pro přechod na distribuované licencování prostřednictvím licenčního serveru lze zařadit několikanásobně větší dohled nad používáním jednotlivých licencí napříč celou organizací.

Motivací ke vzniku této bakalářské práce se tak stala potřeba evidovat historii využití jednotlivých licencí v rámci licenčního serveru na Technické univerzitě v Liberci. Pro tento účel bude vytvořena webová aplikace, která bude simultánně monitorovat dění na licenčním serveru a poskytovat aktuální informace o jeho stavu. Měla by disponovat možností zobrazit platnost spravovaných licencí. Historii využití licenčního serveru lze interpretovat např. formou grafů pro různá časová období. Její nedílnou součástí musí být podpora víceuživatelského prostředí ve spojení s oprávněními.

Při jejím vývoji bude nutné seznámit se s principem fungování licenčního serveru na Fakultě mechatroniky, informatiky a mezioborových studií. Důraz je kladen nejen na samotnou funkčnost, ale také na výkonnost a nízkou náročnost na systémové prostředky serveru. Při návrhu webové aplikace bude nutné zamyslet se nad její bezpečností a neméně důležitou roli sehraje také uživatelská ergonomie. Cíle bude dosaženo v momentě ostrého nasazení na licenční server.

V následující kapitole věnované licenčnímu serveru FLEXIm je uvedena jeho stručná charakteristika a jsou tam popsány jednotlivé úkoly licenčního serveru. Součástí kapitoly se stává vysvětlení principu funkce licenčního serveru včetně popisu dílčích komponent. Kapitola dále obsahuje rozbor možných implementací technologie FLEXIm. Ve třetí kapitole se zabývám rozdíly mezi exitujícími komerčními a open-source webovými aplikacemi. Obě skupiny zde mají své zástupce.

Volba navrženého řešení vychází ze závěrů třetí kapitoly. Čtvrtá kapitola popisuje technologie aplikované při vývoji webové aplikace. Značná část kapitoly je věnována její struktuře včetně popisu jednotlivých modulů aplikační logiky. Kromě nich zahrnuje tato kapitola také rozbor jednotlivých funkcí. Technické detaily v něm uvedené jsou důležité pro orientaci ve zdrojovém kódu. Běžným uživatelům slouží pouze jako manuál k aplikaci.

Minimální požadavky webové aplikace jsou uvedeny v páté kapitole spolu s postupem implementace na licenční server. Součástí kapitoly je také popis dostupných nastavení s ohledem na konfiguraci serveru. Kapitola pojednává o vývoji a obsahuje shrnutí použité testovací metodiky včetně vyhodnocení rychlosti webové aplikace.

## 2 Licenční server FLEXlm

### 2.1 Charakteristika

Softwarová platforma FLEXlm [5] má v oblasti licencování softwaru téměř výsadní postavení. Její první verze byla publikována již v roce 1988 a v současné době je FLEXlm k dispozici ve verzi 11.10.03 pro operační systémy Windows, Mac OS, Linux a mnohé další UNIXové systémy. Licenční řešení FLEXlm spolupracuje s populárním softwarem mnohých známých společností, mezi kterými nechybí společnosti [6] jako ArcGIS (geografické informační systémy), Autodesk (konstrukční software AutoCAD), IBM, Mathworks (simulační prostředí MATLAB), Sun, Synopsys, Texas Instruments, Xilinx (návrhový systém logických obvodů ISE Design Suite) a spousta dalších.

### 2.2 Úkoly licenčního serveru

- správa licenčních souborů
- mechanismus přidělování licencí
- kontrola platnosti jednotlivých licencí
- průběžná kontrola využívání zapůjčených licencí
- rezervace určitého počtu licencí pro konkrétní stanice
- vytváření report logu s provozními informacemi

### 2.3 Princip činnosti

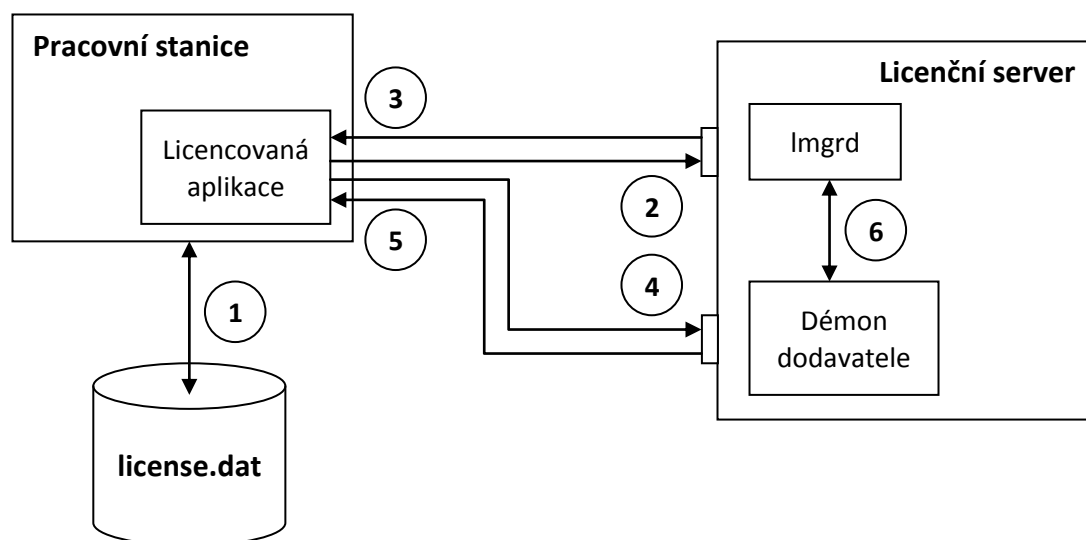
Prostřednictvím plovoucích síťových licencí je umožněno licenčnímu serveru licencovat pouze aktuálně používaný software. Není tedy potřeba licencovat každou instanci instalace softwaru. [6]

Následuje popis mechanismu, ve kterém si klientská aplikace vyžádá příslušnou licenci z licenčního serveru prostřednictvím protokolu TCP/IP.[2] Postup žádosti o vydání plovoucí síťové licence koresponduje s obr. 2.1.

1. Umístění licenčního serveru zjistí aplikace z licenčního souboru, který má k dispozici na svém pevném disku. [3]
2. Na základě požadavku klientské stanice vyhodnotí správce síťových licencí (*lmgrd*) konkrétního démona dodavatele softwaru. Licenční server si na základě IP adresy a doménového názvu počítače ověří identitu klienta.



3. Informace o umístění požadovaného démona dodavatele softwaru předá správce síťových licencí klientské aplikaci. [3]
4. Klientská aplikace naváže spojení s obdrženým démonem dodavatele a odešle mu svoji žádost o zapůjčení licence.[3] Následně si démon dodavatele softwaru na základě aktuálního data a času zkontroluje platnost žádané licence a zjistí počet volných licencí.
5. Po úspěšném ověření všech předpokladů zapůjčí démon dodavatele softwaru plovoucí licenci klientské aplikaci [4] a adekvátně sníží počet dostupných licencí. V opačném případě odešle klientovi zamítavé stanovisko.
6. Démon dodavatele softwaru informuje démona správce síťových licencí o svém rozhodnutí.[4] Správce síťových licencí si obdrženou informaci zaznamená do svého *log souboru*. [4]



Obr. 2.1: Průběh žádosti o vydání plovoucí síťové licence [4]

Démon dodavatele softwaru nadále periodicky kontroluje, zdali klientská aplikace odpovídá a není tak pro ni zbytečně blokována nevyužitá licence.[1] Zároveň však platí, že spuštění další instance licencované aplikace na té samé pracovní stanici nevyžaduje poskytnutí další licence. Metoda, kterou spolu navzájem komunikují démon dodavatele a klientská aplikace, se nazývá signál *heartbeat*. [2] Jestliže nastane situace, kdy je signál *heartbeat* ztracen, pokusí se server o opakované připojení.[2] Po dalším neúspěšném pokusu o spojení, je uživateli zobrazeno chybové hlášení. [2]

V okamžiku ukončení poslední instance licencované aplikace dochází k uvolnění blokové licence. Zapůjčená licence je vrácena do licenčního fondu démona. Démon dodavatele softwaru informuje o této události správce síťových licencí. [4]

## **2.4 Komponenty licenčního serveru**

### **2.4.1 Démon správce licencí (*lmgrd* nebo *lmgrd.exe*)**

Udržuje kontakt s klientskou aplikací a předává spojení démonu dodavatele softwaru.[2] Jeden démon správce licencí může používat pro ověření licence více démonů dodavatelů softwaru.[2] Démon správce licencí má oprávnění spouštět a zastavovat jednotlivé demony dodavatelů softwaru.[2] Zaznamenává výsledky operací, které mu oznamují jednotliví démoni dodavatelů softwaru. [4]

### **2.4.2 Démon dodavatele softwaru (*vendor daemon*)**

Každý výrobce softwaru poskytuje svého démona, který zajišťuje sledování softwaru výrobce.[2] Démon dodavatele monitoruje nejenom licence, ale i počítače, na kterých je používán software výrobce. Uděluje oprávnění k používání spravovaného softwaru.[4] Pravidelně kontroluje využívání licencí na cílových pracovních stanicích.[4] Ukončením démona dodavatele softwaru ztrácejí klientské stanice oprávnění k používání softwaru tohoto výrobce. [2]

### **2.4.3 Licenční soubor (*licence.dat*)**

Obsahuje specifické licenční informace od dodavatele softwaru.[2] Slouží ke konfiguraci licenčního serveru. Z licenčního souboru lze dále určit počet zakoupených licencí pro jednotlivé *features*. Jeho obsah se může změnit v průběhu instalace démona dodavatele. [3]

### **2.4.4 Licencovaná aplikace**

Komunikuje s démonem správce licencí a démonem dodavatele softwaru skrze protokol TCP/IP.[4] Klientská aplikace vyslovuje požadavky a přijímá odpovědi prostřednictvím knihovny poskytovatele technologie FLEXlm. [4]

## **2.5 Způsoby implementace**

### **2.5.1 Model s jedním licenčním serverem**

Správce síťových licencí je nainstalován pouze na jednom fyzickém serveru.[2] Jeden licenční server tak obsahuje všechny licence dostupné na serveru.[2] Licenční server na Fakultě mechatroniky, informatiky a mezioborových studií Technické univerzity v Liberci reprezentuje tento model.

Výhody [2]:

- snadná administrace
- nejnižší náklady na údržbu

Nevýhody [2]:

- selhání serveru způsobí nedostupnost licencovaného softwaru

**2.5.2 Model s distribuovanými licenčními servery**

V tomto modelu jsou licence distribuovány na více fyzických serverech.[2]  
Na každém licenčním serveru je uložen unikátní licenční soubor. [2]

Výhody [2]:

- jednotlivé licenční servery se nutně nemusí nacházet ve stejné podsíti
- selhání jednoho serveru nezpůsobí úplnou nedostupnost poskytovaných služeb
- nahrazení serveru nepřináší nadměrné komplikace

Nevýhody [2]:

- selhání serveru způsobí nedostupnost jím spravovaných licencí
- časově náročnější konfigurace a údržba

**2.5.3 Model s redundantními licenčními servery**

Pro ověřování licenčních souborů se v tomto modelu používají celkem tři licenční servery.[2] Vydávání licencí funguje za předpokladu, že jsou v provozu alespoň dva z nich.[2] Správce síťových licencí je nainstalován na každém serveru, přičemž obsah licenčního souboru je stejný na všech licenčních serverech.[2] Fungování tohoto modelu je podmíněno tím, že se všechny servery musí nacházet ve stejné podsíti. [2]

Výhody [2]:

- selhání jednoho serveru nezpůsobí celkovou nedostupnost služeb

Nevýhody [2]:

- selhání dvou a více serverů znamená nedostupnost licencovaného softwaru
- všechny tři servery se musí nacházet ve stejné podsíti
- model není odolný vůči chybám v síti
- nahrazení jednoho licenčního serveru s sebou nese nutnost rekonfigurace ostatních

## 3 Analýza existujících řešení

### 3.1 Komerční aplikace

Pro licenční optimalizaci existuje celá řada placených aplikací, které nabízejí komplexní správu licenčního serveru. Administrátorům poskytují nástroje pro snazší konfiguraci serveru, sledování systémových informací o stavu serveru nebo disponují možností odeslat e-mail správci v případě nedostupnosti licenčního serveru či vyčerpání konkrétní licence.

Poskytují nástroje pro generování přehledových informací podle celé řady různých hledisek. Tyto reporty se následně uplatňují např. při softwarových auditech nebo jako relevantní podklady při optimalizaci využívání plovoucích síťových licencí. Spousta z nich dále dovoluje monitorovat využívání licencí na konkrétní pracovní stanici nebo umožňuje rezervovat vybrané licence pro konkrétní uživatele.

Předností většiny komerčních aplikací je schopnost monitorovat různé druhy licenčních serverů. Jistou nevýhodou je samozřejmě počáteční investice, kterou však velmi brzy vykompenzují finanční úspory při následném nákupu optimalizovaného počtu licencí.

#### 3.1.1 FlexNet Manager

Tento představitel oficiálního řešení od společnosti Flexera Software je dostupný v několika variantách, které jsou speciálně určeny pro optimalizaci využití softwaru společností Adobe, IBM, Microsoft, SAP a Symantec. Pro srovnání s open-source webovými aplikacemi jsem s ohledem na komplexnost vybral variantu v podobě aplikace FlexNet Manager for Engineering Applications.

Klíčové vlastnosti [8], [9]:

- univerzálnost – kompatibilita s různými informačními systémy
- podpora více druhů licenčních serverů – FLEXlm, LUM, Reprise, Sentinel, ...
- vzdálené ovládání licenčních serverů – spouštění, zastavování, restartování, ...
- centralizovanost – správa licencí prostřednictvím jednotného rozhraní
- otevřenost – data uložena ve zdokumentované databázi
- dynamický přístup – nástroj zpracovává on-line data, nikoli pouze statické pohledy do historie
- informace v reálném čase – hlášení o aktuálním stavu licencí
- velmi dobré reportovací funkce – jádro postaveno na platformě Cognos

- široké spektrum reportů – zahrnuje přednastavené i přizpůsobitelné reporty o stavu licencí, které lze generovat na přání nebo zcela automaticky podle plánu

### 3.1.2 Alternativní komerční řešení

Kromě oficiálního řešení od společnosti Flexera Software samozřejmě existuje několik dalších webových aplikací, na jejichž rozvoji se podílí jiné společnosti. Zpravidla však nenabízí tak široké portfolio funkcí. Jejich výhodou však může být přehlednější uživatelské rozhraní či atraktivnější cenová politika. Pro úplnost následuje seznam několika alternativních řešení z této oblasti.

- LicenseMonitor™<sup>1</sup>
- License Statistics<sup>2</sup>
- OpenLM<sup>3</sup>, ...

## 3.2 Open-source aplikace

Oproti komerčním aplikacím dostupným za úplaty se open-source aplikace vyznačují zejména otevřeným zdrojovým kódem. Všechna analyzovaná řešení jsou šířena pod poměrně benevolentní licencí GPL, která umožňuje libovolné modifikace v kódu. Odvozená díla smí být dále libovolně šířena (dokonce i za úplaty), nicméně nabyvatel licence se zavazuje distribuovat odvozené dílo pouze pod licencí GPL.

Analýzou jednotlivých open-source řešení jsem dospěl k závěru, že jejich implementace a konfigurace vyžaduje mnohdy hlubší znalosti. Komerční aplikace jsou sice nepoměrně rozsáhlejší, nicméně administrátorům je usnadněna jejich implementace prostřednictvím instalačních průvodců. Open-source webové aplikace disponují pouze základní funkcionalitou a nelze od nich očekávat bonusové funkce, kterými se pyšní komerční alternativy. Většina open-source řešení postrádá například správu uživatelů nebo možnost sledovat více serverů.

Jednoznačnou výhodou open-source projektů je široká komunita uživatelů, která posouvá vývoj aplikace kupředu a poměrně pružně reaguje na chyby v ní objevené. Vyskytují se však i projekty, jejichž vývoj zaostává za rozvojem technologií. Počet open-source projektů věnovaných této problematice je z pochopitelných důvodů nižší než počet dostupných komerčních aplikací.

---

<sup>1</sup> <http://www.rtda.com/licensemonitor-monitor-software-license-utilization>

<sup>2</sup> [http://www.x-formation.com/license\\_statistics/index.html](http://www.x-formation.com/license_statistics/index.html)

<sup>3</sup> <http://www.openlm.com/>

### 3.2.1 phpLicenseWatcher<sup>4</sup>

Jednou z možných open-source alternativ je projekt s názvem phpLicenseWatcher, jehož autorem je Vladimír Vuksan. Jádrem celé webové aplikace je napsáno v jazyce *PHP* (verze 4). Webová aplikace používá *PEAR framework* např. pro komunikaci s databází nebo práci s tabulkami. Pro úložiště dat používá dva druhy databází. Dočasná data se ukládají do databáze *MySQL* a pro archivaci dat se používá databáze typu *round-robin*, která si v čase udržuje konstantní velikost. Nástroj *RRDtool* zde zároveň zajišťuje vykreslování grafů. *PEAR framework* i *RRDtool* se musí na platformu *Windows* manuálně doinstalovat a nakonfigurovat. V případě *UNIXových* distribucí bývají již součástí standardní instalace anebo se dají na přání velmi jednoduše přidat.

phpLicenseWatcher umožňuje monitorovat licenční servery *FLEXlm* a *LUM*. Nabízí přehled zapůjčených licencí, poskytuje informace o umístění a aktuálním stavu licenčních serverů. Disponuje možností sledovat platnost zakoupených licencí včetně jejich počtu. V případě nedostatku licencí je zároveň schopný odeslat administrátorovi licenčního serveru hlášení. Mezi jeho další funkce patří možnost zobrazit historii využití konkrétní feature formou grafu.

Vývoj aplikace, zdá se, stagnuje a její vývojáři již pouze opravují nahlášené chyby. Neobjektový návrh aplikace spojený s nutností konfigurovat a udržovat dva typy databází i závislost na *PEAR frameworku* byly jedny z hlavních důvodů, které rozhodly ve prospěch následujícího projektu.

### 3.2.2 flexmonitor<sup>5</sup>

Autorem tohoto nadějněho projektu je programátor s přezdívkou „killerxcana“. Projekt je prozatím ve fázi testování, nicméně verze 3.0 z 2. ledna 2011 doznala několika zásadních změn oproti verzím předcházejícím. Mimo jiné byl kompletně přepsán její zdrojový kód za účelem kompatibility s architektonickým vzorem *MVC* (*Model-View-Controller*). Úprav se dočkala také samotná databázová vrstva, jejíž struktura byla optimalizována za účelem zvýšení efektivity vykreslování grafů. Z původních dvou naplánovaných úloh zbyla v nové verzi pouze jediná, apod. V následující kapitole detailně popisují současnou podobu aplikace, která odráží potřeby Fakulty mechatroniky, informatiky a mezioborových studií.

---

<sup>4</sup> <http://phplicensewatch.sourceforge.net/>

<sup>5</sup> <http://flexmonitor.sourceforge.net/>

## 4 Webová aplikace

### 4.1 Použité technologie

#### 4.1.1 PHP

Jádro celé webové aplikace je naprogramováno v jazyce PHP 5, který s sebou přinesl kromě řady vylepšení také dlouho očekávanou podporu objektově orientovaného programování. Samotný jazyk je dostupný pod licencí PHP license, která se vztahuje na redistribuci zdrojových a binárních kódů PHP a na vytváření jeho odvozenin.[12] PHP skripty jsou vykonávány na straně serveru na rozdíl např. od JavaScriptu.

#### 4.1.2 JpGraph

Tato objektově orientovaná grafická knihovna je naprogramovaná také v jazyce PHP. K dispozici je ve dvou verzích [13], které se liší množstvím nabízených funkcí. Verze pro nekomerční použití je chráněna licencí QPL 1.0<sup>6</sup>. Komerční verze Professional nabízí navíc několik druhů grafů, práci s čárovými nebo QR kódy, vizualizaci tabulek, zahrnuje tříměsíční e-mailovou podporu ze strany vývojářů, apod. JpGraph implementuje vlastní kešovací systém (viz *cache*), nabízí několik druhů grafických formátů obrázků, vyhlazování grafů, podporuje ukládání grafů do souborů, vyznačuje se obsáhlou dokumentací, apod.

Hlavním argumentem pro volbu této knihovny bylo to, že grafy jsou generovány na straně serveru a v souvislosti s jejich vykreslováním tak nejsou kladeny žádné další požadavky na softwarové vybavení klienta. Pro jejich zobrazení není zapotřebí ani JavaScript. S tímto přístupem jsou sice spojeny vyšší paměťové a výpočetní nároky kladené na server, nicméně se předpokládá, že webová aplikace bude používána omezeným počtem autorizovaných uživatelů. Vyšší požadavky jsou částečně kompenzovány kešováním grafů. Více informací se o technice kešování grafů dozvíte v kapitole 4.5.3.

#### 4.1.3 MySQL

Databázový systém MySQL lze získat s licencí GPL nebo s komerční licencí.[12] Program distribuovaný spolu s MySQL databází musí být šířen také pod licencí GPL.[12] Zmíněnou nevýhodou netrpí komerční licence MySQL. [12]

---

<sup>6</sup> <http://www.opensource.org/licenses/qtpl.php>

Struktura MySQL databáze byla navržena s ohledem na výkon a zejména nízkou náročnost na systémové prostředky licenčního serveru. Z tohoto důvodu bylo zvoleno úložiště MyISAM, které díky své jednoduchosti vyniká zejména při větším množství dotazů. Databáze se skládá z celkem sedmi tabulek, přičemž vzájemné vztahy mezi entitami popisuje ER diagram (viz Příloha H). Tabulka `servers` slouží pro uchování informací o monitorovaných licenčních serverech. Tabulka `features` obsahuje seznam všech features. Dostupné demony dodavatelů softwaru udržuje tabulka `products`. Záznamy v tabulce `ports` reprezentují čísla portů, na kterých naslouchají jednotliví démoni. Řádky v tabulce `licenses` slouží k propojení licenčního serveru s konkrétním démonem dodavatele a jeho portem. Každý den se ukládá počet zakoupených licencí u každé feature včetně její příslušnosti k démonu dodavatele do tabulky `licenses_available`. V klíčové tabulce `licenses_usage` se uchovávají informace o využití jednotlivých features v čase. Vedle identifikátoru konkrétní feature disponuje každý záznam časovým razítkem, počtem aktivních uživatelů a příslušností feature k jejímu démonu. V tabulce `users` lze nalézt informace o registrovaných uživateli. Jednotlivé sloupce této tabulky představují jeho uživatelské jméno, otisk heslové fráze spolu se „solí“ (viz kapitola 4.5.5), registrovanou e-mailovou adresu, jméno a příjmení, cookie token (volitelně), identifikátor session relace (volitelně), IP adresu a datum a čas posledního aktivity (volitelně).

#### 4.1.4 jQuery

Webová aplikace používá tento populární *JavaScriptový* framework pro ošetření a validaci vstupních dat. jQuery se v ní dále uplatňuje při požadavku na změnu období v rámci historie využití. Požadavek je zpracován prostřednictvím této knihovny bez nutnosti znovu načítat celou stránku. Celá aplikace je však důmyslně navržena i pro uživatele se zakázaným JavaScriptem.

jQuery se vyznačuje zejména snadnou rozšiřitelností. Dynamickou práci s tabulkami zprostředkovává plugin `TableSorter`<sup>7</sup>. Vzhledem k odlišnosti českého jazykového prostředí od anglického jsem rozšířil jeho kód o metodu, která zajišťuje správné řazení českého formátu data. Řazení podle více sloupců zároveň se provádí za stálého držení klávesy Shift. Plugin `Validation`<sup>8</sup> zaručuje validaci vstupů na straně

---

<sup>7</sup> <http://tablesorter.com/docs/>

<sup>8</sup> <http://bassistance.de/jquery-plugins/jquery-plugin-validation/>



klienta. Rovněž vyniká snadnou rozšiřitelností a dokonce disponuje možností definovat vlastní informační hlášení. Přidané validační metody obsahuje skript `jquery.validate.min.additional-methods.js` v adresáři `includes/scripts/` a lokalizovaná chybová hlášení jsou prezentována v souboru `jquery.validate.min.messages_cs.js`, který se nachází ve stejném adresáři.

#### 4.1.5 XHTML

Všechny části webové aplikace splňují požadavky konsorcia W3C na jejich validitu. Webové stránky dodržují specifikace modularizovaného jazyka XHTML 1.1. Testem na syntaktickou správnost prošly rovněž kaskádové styly. Výchozím kódováním používaným v rámci všech modulů i v databázovém systému je UTF-8.

#### 4.1.6 Vzhled aplikace

Rozložení a styl uživatelského prostředí webové aplikace jsou definovány pomocí kaskádových stylů, které jsou reprezentovány souborem `style.css` v adresáři `includes`. Hlavní menu aplikace vychází z volně dostupného menu Blue Tabbed Drop Down<sup>9</sup>. Stavové ikony byly převzaty z doplňkového balíčku ikon DarkGlass<sup>10</sup> určených pro desktopové prostředí KDE 3.x. Balíček je distribuován pod licencí GPL a jeho autorem je Alessandro Rei. Ostatní ikonky pochází z balíčku Silk Icons<sup>11</sup> od Marka Jamese, který je uvolnil pod licencí Creative Commons Attribution 3.0 License.

### 4.2 Dokumentace v kódu

Většina metod je uvozena dokumentačním komentářem, který zahrnuje její výstižný popis, modifikátor přístupu, předávané parametry (volitelně) a samozřejmě podává informaci o tom, co metoda vrací. Ukázku formátu dokumentačního komentáře naleznete např. u šifrovací funkce (viz Příloha B). Pro zpřehlednění kódu slouží technika odsazování v kombinaci s jednořádkovými komentáři. Jednořádkové komentáře vysvětlují méně srozumitelné části kódu nebo popisují parametry funkcí jazyka PHP, které přebírají větší množství argumentů.

---

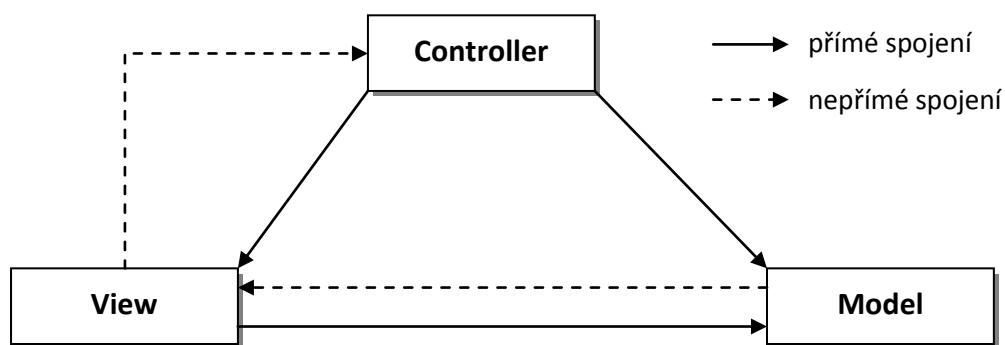
<sup>9</sup> <http://www.cssmenuaker.com/>

<sup>10</sup> [http://kde-look.org/content/show.php/DarkGlass\\_Reworked?content=67902](http://kde-look.org/content/show.php/DarkGlass_Reworked?content=67902)

<sup>11</sup> <http://famfamfam.com/lab/icons/silk/>

### 4.3 Architektonický vzor

Struktura aplikace byla navržena s ohledem na budoucí snadnou rozšiřitelnost. Začleněním návrhového vzoru MVC bylo dosaženo oddělení datového modelu od řídicí logiky aplikace a uživatelského rozhraní. Tento přístup má výhodu v tom, že umožňuje snadno přidávat, upravovat nebo nahrazovat jednotlivé pohledy, které sdílí společný model. Jednoduchá struktura (viz obr. 4.1) definovaná tímto architektonickým vzorem má pozitivní vliv na výkon. MVC se dále vyznačuje nízkou provozní režii.



Obr. 4.1: Principiální schéma architektonického vzoru Model-View-Controller

Komponenty řadiče jsou uloženy v adresáři `controller`, komponenty modelu se nacházejí v adresáři `model` a pohledy na data jsou shromažďovány do adresáře `views`. Vlastní aplikační logika se skládá z pěti hlavních modulů, které spojují jednotlivé komponenty vzoru MVC. Zmiňované moduly jsou soustředěny v adresáři `application`. Rozbor jejich funkcionality naleznete v kapitole 4.4.

### 4.4 Aplikační logika

#### 4.4.1 `controller_base.class.php`

Součástí aplikační logiky je abstraktní třída `baseController`, která inicializuje objekt `registry`. Tento objekt je přístupný všem odvozeným třídám pro deklaraci vlastních klíčů nebo jim umožňuje přistoupit k hodnotám v něm uložených. Abstraktní třída dále předepisuje abstraktní metodu `index()`, která musí být implementována ve všech odvozených třídách.

#### 4.4.2 `registry.class.php`

Tělem skriptu je třída `Registry`, která zajišťuje uložení hodnoty do objektu registru prostřednictvím metody `__set($index, $value)`. Třída dále implementuje metodu `__get($index)`, která vrací požadovanou hodnotu z registru.

#### 4.4.3 `router.class.php`

Skript obsahuje třídu `Router`, která zajišťuje zpracování požadované adresy. Metoda `setPath($path)` nastavuje cestu k adresáři `controller` při požadavku na načtení stránky. Metoda `getController()`, zpracuje požadovanou adresu, určí odpovídající řadič a uloží akci, která se má vykonat. V těle metody se také ověřuje oprávnění uživatele vykonat požadovanou akci. Metoda ukládá vybraný řadič do registru za účelem zvýraznění pozice v menu. Poslední metoda, která se nachází v těle této třídy, je metoda `loader()`. Tato metoda zkontroluje existenci získaného řadiče, vytvoří jeho instanci a předá mu objekt `registry`. Ověří vykonatelnost získaného požadavku a předá ho řadiči i s případnými argumenty.

#### 4.4.4 `template.class.php`

Tento skript reprezentuje třída `Template`, která přebírá výsledky od řadiče. Prostřednictvím metody `show($name)` sestavuje šablonu, která se skládá z hlavičky portálu, menu (volitelně), hlášení, zvoleného pohledu prezentujícího výsledky operace a záhlaví portálu. Metoda zároveň zpřístupňuje všem komponentám šablony informace o aktuálně přihlášeném uživateli, konfiguraci této webové aplikace a mnohé další parametry potřebné například pro dynamické generování menu.

#### 4.4.5 `user.class.php`

Tento modul představuje třída `User`, která provádí autentizaci uživatele a vše s tím spojené. Třída je kompatibilní s návrhovým vzorem *Singleton*, který zaručuje vytvoření pouze jedné instance třídy a poskytuje k ní globální přístup. Instance této třídy je vytvořena při každém požadavku skriptem `init.php`, který se nachází v adresáři `includes`, a následně je globálně zpřístupněna prostřednictvím objektu `registry`.

Autentizace uživatele spočívá v několika krocích, které se mírně liší v závislosti na typu přihlášení. V prvním případě se ověřuje existence *session* identifikátoru relace, kdy dochází ke kontrole vzájemné příslušnosti této relace s uživatelským jménem a IP adresou klienta. Uživatelské jméno se předává pomocí *session* proměnné, přičemž IP adresu klienta zjišťuje server. Zároveň se kontroluje, zdali nevypršel časový limit pro automatické odhlášení při neaktivitě definovaný v konfiguračním souboru `config.php`, který se nachází v adresáři `conf`.

Ve druhém případě se ověřuje platnost přihlašovacích *cookies*, které indikují trvalé přihlášení (stanovené na dobu šesti měsíců). Uživatel je identifikován podle uživatelského jména uloženého v první cookie. Současně s ním je kontrolována shoda unikátního přihlašovacího tokenu (získaného z druhé cookie) oproti záznamu v databázi. V případě shody je vygenerován náhodný identifikátor relace, který se vloží do databáze spolu s IP adresou klienta, přičemž se regeneruje čas posledního přístupu. Následně se vytvoří session proměnné obsahující relaci a uživatelské jméno z toho důvodu, aby se při každém požadavku nemusely kontrolovat cookies. Autentizaci poté zajišťuje mechanismus session až do doby, než vyprší platnost session relace. Po jejím skončení se jednorázově provede autentizace pomocí cookies, další autentizace probíhají skrze session mechanismus z prvního případu. Celý algoritmus se nadále periodicky opakuje.

## 4.5 Funkce aplikace

Každá položka v hlavním menu aplikace má svůj samostatný řadič, který poskytuje metody pro jednotlivé funkce. Obsluhu vybraného požadavku zabezpečuje příslušná komponenta řadiče, která je umístěna v adresáři `controller`.

### 4.5.1 Úvodní přehledová strana

Načtení informací zobrazených na úvodní straně zajišťuje řadič `index.php` ve spolupráci s modely `db.class.php` a `licengine.class.php`. Úvodní přehledová strana představuje výchozí bod portálu, na který je uživatel přesměrován po dokončení standardních procedur (přihlášení, odhlášení, ...) nebo v případě problémů (chyba serveru, skriptu, nedostupnost stránky, ...). Na požádání je přístupná po kliknutí na odkaz **Přehled** v hlavním menu nebo následováním odkazu v levém horním rohu každé stránky.

Podoba úvodní přehledové strany se mění v závislosti na výsledku autentizace uživatele. Nepřihlášený uživatel je pouze informován o aktuálním stavu licenčního serveru. Přihlášenému uživateli je navíc zobrazen výčet sledovaných serverů doplněný informacemi o jejich umístění spolu s verzí instalovaného manažera síťových licencí. V případě, že byla pro dnešní den v rámci všech licenčních serverů zapůjčena alespoň jedna plovoucí síťová licence, bude v pravé části úvodní stránky zobrazena tabulka maximálního využití features, kterým byla dnes zapůjčena alespoň jedna licence. Features jsou v tabulce implicitně řazeny sestupně podle nejvyššího využití. Kliknutím na konkrétní feature bude zobrazena historie jejího využití. Umístění

tabulky se dynamicky mění v závislosti na šířce viewportu (tj. viditelná část webové stránky). Pod tabulkou monitorovaných serverů se nachází graf denního využití nejvíce využívané feature v rámci všech licenčních serverů. Podobu úvodní přehledové strany přihlášeného uživatele zobrazuje Příloha C.

#### 4.5.2 Aktuální využití

Zobrazení informací o aktuálním využití licenčních serverů zprostředkovává řadič `realtime.php` ve spolupráci s modelem `db.class.php`.

Následováním odkazu **Aktuální využití** v hlavním menu aplikace se zobrazí statistika aktuálního využití licenčních serverů. Pro každý server je zobrazeno celkové procentuální využití a tabulka démonů dodavatelů softwaru spolu s informacemi o jejich umístění, aktuálním stavu, nejvyšším využití konkrétní feature(s), celkovém využití včetně verze démona dodavatele. Snímek obrazovky zachycující statistiku aktuálního využití licenčního serveru je k nahlédnutí jako Příloha D.

Detailní statistiku využití lze zobrazit kliknutím na jméno démona. Stejného výsledku lze dosáhnout výběrem příslušného démona z nabídky zobrazené po najetí na odkaz **Aktuální využití** v hlavním menu aplikace. Buňky ve sloupci **Nejvyšší využití** nabízejí informaci o nejvyžívanější feature formou popisku, který se zobrazí po najetí na příslušnou buňku. Popisek může obsahovat seznam features vzájemně oddělených čárkami v případě, kdy mají stejné maximální procentuální využití. Popisek nebude pochopitelně zobrazen při nulovém procentuálním využití démona.

Uživatelé se zapnutým JavaScriptem ocení možnost dynamicky řadit data podle jednotlivých sloupců tabulky. Data v tabulce jsou implicitně řazena podle následujícího klíče: sestupně podle nejvyššího a celkového využití, poté abecedně podle názvu démona. Kliknutím kdekoli v příslušném řádku se zobrazí detailní statistika aktuálního využití démona.

Detailní statistika poskytuje souhrn všech features zvoleného démona dodavatele softwaru. Pro každou feature je uveden celkový počet zakoupených licencí, sloupec **Rezervováno** představuje počet zapůjčených licencí a buňky ve sloupci **Volné** představují procentuální vyjádření dostupných licencí. Výchozí řazení sloupců je následující: vzestupně podle sloupce **Volné**, sestupně podle sloupce **Rezervováno** a abecedně podle názvu feature. Kliknutím na název feature se uživatel přesune na historii jejího využití. Informace o zapůjčených licencích se zobrazuje pod danou feature. Uživatel se zapnutým JavaScriptem rozbálí podrobné informace

kliknutím na počet zapůjčených licencí. V prvním sloupci se nachází souhrnná informace o klientovi, druhý sloupec zobrazuje délku rezervace licence ve formátu *H.m:s*.

#### 4.5.3 Historie využití

Řadiče `monitor.php` a `graph.php` ve spolupráci s modely `db.class.php` a `licengine.class.php` zodpovídají za zobrazení historie využití démonů dodavatelů softwaru i jednotlivých features. Historie využití je zobrazena pomocí grafů pro různá časová období.

Na rozcestník obsahující jednotlivé features rozdělené podle svých démonů se uživatel dostane po kliknutí na odkaz **Historie využití** v hlavním menu aplikace. Seznam démonů dodavatelů softwaru je dále seskupen podle příslušnosti ke konkrétnímu licenčnímu serveru. Uživatel se zapnutým JavaScriptem kliknutím na jméno démona rozbálí/sbalí seznam spravovaných features. Uživateli bez podpory JavaScriptu je pro každý server zobrazena tabulka, ve které jednotlivé řádky s přepínačem odpovídají konkrétnímu démonu dodavatele. Vpravo vedle jména démona se zobrazuje vysouvací nabídka, která obsahuje seznam poskytovaných features. Po vybrání démona dodavatele a příslušné feature se volba potvrzuje tlačítkem **Zobrazit** umístěným pod tabulkou.

Historie využití konkrétní feature je zobrazena pomocí grafu zapůjčených licencí v závislosti na čase. Měřítka na ose X se přizpůsobuje délce konkrétního časového období. Měřítka na ose Y je určeno počtem zakoupených licencí. Popisky na ose X se řídí zvoleným časovým obdobím. Historii využití si lze zobrazit pro dnešní den, poslední týden, měsíc zpátky, rok nazpět nebo si lze nechat vykreslit všechna dostupná data. Volba všechna data znamená zobrazení nejvyššího využití konkrétní feature nebo démona postupně pro všechny zaznamenané dny. Zobrazení kompletní historie využití by jinak kladlo značné nároky na výpočetní výkon serveru a v případě velkého množství záznamů by snižovalo celkovou čitelnost grafu. Výchozím zobrazením je denní využití, nicméně pro uživatele se zapnutým JavaScriptem je nad příslušným grafem k dispozici přepínač období, který slouží pro přepínání grafů bez nutnosti načítání celé stránky. Uživatelé bez podpory JavaScriptu mají kromě přepínačů k dispozici také tlačítko **Zobrazit**, jehož aktivací dojde k překreslení grafu v závislosti na zvoleném období. Dalším možným způsobem, jak si zvolit období, je přidat parametr formou anglického pojmenování

období za číslo feature do adresy URL<sup>12</sup>. Možné kombinace jsou: „day”, „week”, „month”, „year” a „all”. Ukázku zobrazení historie využití licencí programu MATLAB naleznete pod označením Příloha E.

Historii využití jednotlivých démonů dodavatelů lze zobrazit výběrem z nabídky zobrazené po najetí na položku **Historie využití** v hlavním menu aplikace. Všechny výše uvedené skutečnosti platí také pro tento modul s několika málo změnami. Jednou z nich je měřítko osy Y, které se nyní přizpůsobuje maximálnímu počtu ze součtu zapůjčených licencí v daný okamžik. Graf tedy zobrazuje součet zapůjčených licencí daného démona dodavatele v závislosti na konkrétním čase.

Pro snížení zátěže kladené na server a databázi při vykreslování grafů implementuje knihovna JpGraph svůj vlastní kešovací systém. Při prvním požadavku na vykreslení konkrétního grafu se kromě jeho odeslání do okna prohlížeče současně vytvoří v nastaveném dočasném adresáři na serveru jeho kopie. Platnost této kopie lze ovlivnit nastavením proměnné `$timeout` v minutách pro každé období zvlášť. Časový limit lze nastavit separátně pro features i demony dodavatele softwaru v řadiči `graph.php`, který obsahuje příslušné metody `image_feature($args)` a `image_daemon($args)`. Při každém požadavku na vykreslení grafu se nejprve zjišťuje existence kopie v zásobníku. Jestliže je nalezena, ověří se její aktuálnost a odešle se klientovi, pokud ještě nevypřela její platnost. V opačném případě se získají aktuální data z databáze a soubor s grafem se přepíše. Pozitivním důsledkem je zrychlení vyřízení požadavku na získání grafu.

#### 4.5.4 Platnost features

Zjištění platnosti features zajišťuje řadič `expire.php` ve spolupráci s modelem `db.class.php`.

Vybráním položky **Platnost licencí** v hlavním menu aplikace získává uživatel souhrnné informace o všech zakoupených licencích a jejich stavu. Pro každý licenční server je zobrazena samostatná tabulka. Řádky tabulky reprezentují jednotliví démoni dodavatelů softwaru spolu s informacemi o jejich umístění, počtu spravovaných features, množství features s omezenou platností a nechybí ani informace o nejbližším datu expirace licence včetně počtu expirovaných licencí. Tabulka se souhrnem platnosti licencí je zachycena na snímku Příloha F.

<sup>12</sup> Uniform Resource Locator jednoznačně specifikuje umístění zdroje v internetu. [7]

Kliknutím na jméno démona se zobrazí podrobnější informace o monitorovaných licencích. Stejného výsledku lze dosáhnout výběrem příslušného démona z nabídky zobrazené po najetí na odkaz **Platnost licencí** v hlavním menu aplikace. Buňky ve sloupcích **Nejbližší expirace** a **Po splatnosti** nabízejí doplňující informaci o features, kterých se to týká, formou popisku, který se zobrazí po najetí na příslušnou buňku. Popisek samozřejmě nebude zobrazen u démonů, kteří nedisponují žádnou licencí s omezenou platností.

Uživatelé se zapnutým JavaScriptem mají umožněno manipulovat s výchozím řazením sloupců a přizpůsobit si tak zobrazení dat v tabulce. Řazení dat je implicitně nastaveno v tomto pořadí: podle počtu licencí po splatnosti sestupně, podle data nejbližší expirace, dále sestupně podle počtu features s omezenou platností a posledním kritériem je abecedně řazený sloupec se jménem démona. Kliknutím kdekoli na řádku se zobrazí podrobnější informace o licencích spravovaných příslušným démonem dodavatele.

Podrobný souhrn licencí démona dodavatele softwaru obsahuje vedle názvu feature také příslušný počet zakoupených licencí včetně informace o tom, kdy vyprší. V případě, že je v buňce vyjadřující počet zakoupených licencí uvedeno „9999“, znamená to, že daná feature disponuje neomezeným počtem licencí, které nikdy nevyprší. Číslo v posledním sloupci vyjadřuje počet dní, které zbývají do skončení platnosti příslušných licencí. Po jejich expiraci se zde zobrazuje text „vypršeno“; u doživotních licencí je buňka proškrtnutá.

Řazení dat ve výpisu je opět umožněno pouze uživatelům, kteří mají disponují podporou JavaScriptu. Záznamy jsou implicitně řazeny vzestupně podle počtu dní, které zbývají do vypršení platnosti a dále abecedně podle názvu feature.

#### 4.5.5 Administrace

Tato sekce je přístupná pouze registrovaným uživatelům s administrátorskými právy. Administrační modul obsluhuje řadič `admin.php` ve spolupráci s modelem `db.class.php`.

Správce licenčního serveru má zde možnost přidávat nové licenční servery, odstraňovat ty stávající nebo konfigurovat jejich nastavení. Zároveň se zde zobrazuje aktuální velikost databáze. Rozložení administračního rozhraní představuje Příloha G.

Proces založení a aktivace uživatelského účtu probíhá v několika krocích. Správce licenčního serveru nejprve vyplní registrační formulář, který je dostupný



z administrační sekce. V rámci uživatelského účtu se uchovává unikátní uživatelské jméno a e-mailová adresa, rozsah oprávnění, jméno a příjmení. Uživatelské jméno se dále používá pro přihlašování, e-mailová adresa slouží k dokončení registrace, z pohledu oprávnění se rozlišují uživatelské role – administrátor a běžný uživatel. Jméno a příjmení přihlášeného uživatele se zobrazuje v pravém horním rohu aplikace.

Po kontrole unikátnosti zadaného uživatelského jména se ověřuje validita e-mailové adresy podle platných RFC<sup>13</sup> dokumentů včetně kontroly existence DNS<sup>14</sup> záznamu pro doménovou část zadané e-mailové adresy. Metoda `validateEmail($email)`, která potřebný test provádí, se nachází v souboru `users.class.php` (viz Příloha A). Poté se vygeneruje šestimístný PIN kód složený z náhodné kombinace číslic a písmen. Administrátor předá kód uživateli, který ho bude následně potřebovat při prvním přihlášení do aplikace. Součástí dvoucestného ověřování je také potvrzovací e-mail odeslaný na registrovanou e-mailovou adresu. Obsahem e-mailu je odkaz na úvodní stránku této webové aplikace doplněný o náhodně generovaný identifikátor. Kombinace PIN kódu a identifikátoru je v databázi uložena v zabezpečené formě použitím hashovací funkce SHA-2 doplněné o náhodně generovanou sekvenci 64 znaků označovanou jako „sůl“. Tento přístup slouží jako ochrana proti pokusu o prolomení hesla při útoku hrubou silou ve spojení s „duhovými“ tabulkami. Ukázka použité šifrovací funkce viz Příloha B.

První přihlášení se provádí následováním odkazu z obdrženého e-mailu, následným vyplněním uživatelského jména a osobního PIN kódu. Po stisku tlačítka **Přihlásit** se ze zadané kombinace PIN kódu a unikátního identifikátoru vytvoří otisk použitím šifrovací funkce (viz Příloha B). Z databáze se pro zadané uživatelské jméno získá hash, který se očistí od přidané „soli“. Přihlašování je dokončeno v případě shody obou otisků. Uživateli je následně zobrazen formulář pro nastavení nového hesla. Minimální délka hesla je stanovena na šest znaků a nové heslo je v databázi uloženo stejným způsobem jako při registraci uživatele s tím rozdílem, že místo PIN kódu a náhodného identifikátoru se šifruje a ukládá pouze zadaná heslová fráze. Nastavením heslové fráze tak uživatel ztrácí možnost přihlášení pomocí PIN kódu a unikátního identifikátoru. Pokud by si uživatel heslo z jakéhokoli důvodu nenastavil,

---

<sup>13</sup> Request For Comments představují oficiální nezávazné technické dokumenty, které popisují fungování internetu a definují některé síťové protokoly.

<sup>14</sup> Domain Name System je hierarchicky uspořádaná struktura doménových jmen, která je realizována systémem DNS serverů a komunikačním protokolem DNS. Slouží ke vzájemnému převodu IP adres na doménová jména a zpět.

opětovné přihlášení je mu umožněno zopakováním předchozího postupu, který předpokládá znalost PIN kódu a speciálního odkazu z potvrzovacího e-mailu.

#### 4.5.6 Přihlašování, odhlašování a změna hesla uživatele

Registrovaní uživatelé se přihlašují do webové aplikace svým uživatelským jménem a zvoleným heslem. Ověření správnosti zadaných údajů se provádí stejným způsobem jako při prvním přihlášení. Podrobnosti včetně procesu autentizace uživatele jsou detailně popsány v oddílu 4.5.5. Úroveň oprávnění přihlášeného uživatele indikuje ikona vedle jeho jména. Změnu heslové fráze má uživatel možnost provést odesláním příslušného formuláře dostupného pod odkazem ***změnit heslo***. Odhlašování probíhá stiskem tlačítka ***Odhlásit***, které se nachází v pravém horním rohu aplikace. Při procesu odhlašování se kontroluje platnost konkrétní session relace ve spojení s IP adresou klienta. Identifikátor relace se zneplatňuje na straně klienta i databáze. V rámci trvalého přihlášení se likvidují také nastavené přihlašovací cookies.

## 5 Implementace na licenční server a měření výkonu

### 5.1 Požadavky

- webový server Apache
- PHP 5.3 a vyšší
- MySQL databáze

#### 5.1.1 Apache

Webová aplikace vyžaduje extenzi `mod_rewrite`, která se používá při přepisování URL adres. Spolupráce se souborem `.htaccess` je zajištěna nastavením direktivy `AllowOverride` na hodnotu `all` v konfiguračním souboru `httpd.conf` webového serveru Apache. Ve stejném konfiguračním souboru je také potřeba nastavit volbu `Options FollowSymLinks` minimálně pro adresář s aplikací. Textový soubor `.htaccess` upravuje některé vlastnosti webového serveru. V případě neplatné cesty např. předává její cíl k ošetření kořenovému souboru `index.php` formou parametru. V souvislosti s bezpečností této webové aplikace dále upravuje chování `session` a spolu se souborem `.htpasswd` chrání vybrané adresáře před neoprávněným přístupem. V adresářích `conf`, `includes`, `logs` a `tmp` se proto nachází skrytý soubor `.htaccess`, ve kterém je nutné v rámci konfigurační direktivy `AuthUserFile` specifikovat absolutní cestu k souboru `.htpasswd`. Obsahem souboru `.htpasswd` je seznam přístupových údajů ve tvaru uspořádané dvojice `jméno:heslo`. Heslo je z bezpečnostních důvodů šifrované algoritmem SHA-1. Pro jeho zašifrování doporučuji použít on-line nástroj na webových stránkách [AskApache.com](http://www.askapache.com)<sup>15</sup>.

#### 5.1.2 PHP

Modul webového serveru jazyka PHP musí podporovat extenze pro práci s databází MySQL a knihovnu Zlib, která je nutná pro podporu komprese v rámci protokolu HTTP/1.1 používanou webovým serverem Apache. Odesílání registračních e-mailů přes webovou aplikaci vyžaduje správné nastavení cesty k programu `sendmail` v konfiguračním souboru `php.ini`. V sekci `[mail function]` je proto k dispozici direktiva `sendmail_path`. Výchozí cesta včetně příslušných parametrů je na většině UNIXových systémů `/usr/sbin/sendmail -t -i`. Z důvodu

---

<sup>15</sup> <http://www.askapache.com/online-tools/htpasswd-generator>

bezpečnosti doporučuji zakázat odesílání jména skriptu v hlavičce každého e-mailu nastavením konfigurační direktivy `mail.add_x_header` na `off`. Dále je nutné zkontrolovat přítomnost extenze `mcrypt`, která je v aplikaci používána pro vytvoření náhodného inicializačního vektoru při vytváření „soli“ k heslové frázi.

### 5.1.3 JpGraph

Vyžaduje povolení extenze GD v konfiguračním souboru `php.ini`. Podporu modulu GD lze ověřit příkazem `phpinfo()`. Dále je nutné s ohledem na nastavení serveru upravit konfigurační direktivy v souboru `jpg-config.inc.php`, který se nachází v adresáři `includes/jpgraph`. Důležitá je zejména konstanta `CACHE_DIR`, která definuje cestu k adresáři, kam budou ukládány generované grafy připravené pro budoucí použití. Implicitní cesty pro operační systémy UNIX i Windows jsou uvedeny v dokumentačním komentáři nad direktivou. Zároveň je potřeba schválit použití kešovacího systému nastavením konfiguračních direktiv `USE_CACHE` a `READ_CACHE` na `true`.

### 5.1.4 MySQL

Z bezpečnostních důvodů doporučuji nastavit heslo uživateli `root` příkazem `mysqladmin -u root -password <nové_heslo>`. Proces nastavení MySQL databáze používané touto webovou aplikací začíná příkazem:

```
mysqladmin -u root -p create <jméno_databáze>. Ještě před samotným  
založením prázdné databáze však budete vyzváni k zadání přístupového hesla  
uživatele root. V dalším kroku se přihlásíte pod uživatelem root příkazem:  
mysql -u db_user -p. Nyní je potřeba založit nového uživatele a udělit mu  
oprávnění pouze na nově vytvořenou databázi. Pro tento účel se používá příkaz:  
GRANT ALL PRIVILEGES ON <jméno_databáze>.*  
TO <jméno_uživatele>@localhost IDENTIFIED BY <nové_heslo>;.
```

V případě úspěchu obdržíte odpověď: `Query OK, 0 rows affected`. Načtení tabulek privilegií zajišťuje v MySQL příkaz: `FLUSH PRIVILEGES;`.

Databázovou strukturu definuje SQL skript `database.sql`, který se nachází v adresáři `conf`. Název databáze lze specifikovat na řádcích `CREATE database <jméno_databáze>... a USE <jméno_databáze>`. Přihlašovací údaje nutné pro spojení webové aplikace s MySQL databází se nastavují v konfiguračním souboru `config.php` v adresáři `conf`. Import databázové struktury včetně všech entit,

atributů, indexů i komentářů lze zajistit příkazem: `mysql -u <jméno_uživatele> -p <jméno_databáze> < cesta_k_souboru_database.sql>`.

Ukončení práce s konzolí MySQL se provádí příkazem: `\q`.

## 5.2 Konfigurace aplikace

V souboru `config.php` jsou k dispozici následující konfigurační direktivy:

- `base_url` – označuje výchozí URL adresu
- `base_ip` – určuje IP adresu stanice, ze které se bude vykonávat skript `licenses.php`; jejím nastavením se zamezí spuštění skriptu mimo server
- `app_title` – umožňuje nastavit titulek webové aplikace
- `logfile` – specifikuje relativní cestu k log souboru aplikace; doporučuji vytvořit adresář `logs` v kořenovém adresáři aplikace
- `regfile` – nastavuje relativní cestu k textovému souboru s uživatelskými PIN kódy; z bezpečnostního hlediska jej doporučuji ponechat v adresáři `logs` u aplikace, kde mu poskytuje ochranu soubor `.htaccess`
- `admin` – představuje kontaktní e-mailovou adresu správce licenčního serveru odkazovanou v registračním e-mailu
- `checktimelimit` – určuje časový limit před automatickým odhlášením v případě nečinnosti; výchozí hodnota 1 800 s představuje kompromis mezi bezpečností a obtěžováním uživatele přihlášeného pomocí `session`
- `lmutil_loc` – definuje absolutní cestu k souboru nástrojů `lmutil` poskytovatele technologie FLEXlm; výchozí cesta v UNIXových operačních systémech je `/usr/local/flexlm/bin/lmutil`
- `hostname` – označuje umístění databáze
- `username` – specifikuje uživatelské jméno pro přístup k databázi
- `password` – nastavuje heslovou frázi pro přístup k databázi
- `database` – specifikuje konkrétní databázi příslušející této webové aplikaci.

## 5.3 Dokončení instalace

V této fázi je konfigurace závislostí úspěšně dokončena. Deaktivace autentizačních mechanismů a povolení rozšířených chybových hlášení se provádí nastavením konfigurační direktivy `__DEBUG` na hodnotu `true` v souboru `index.php`. Po přesunu na úvodní stránku této webové aplikace definované konfigurační

direktivou `base_url` se přesuňte do administrační sekce, do které se dostanete zadáním názvu příslušného řadiče (`admin`) na konec URL adresy. Vyplňte požadované informace o licenčním serveru v příslušném formuláři, který zobrazíte stisknutím tlačítka **Přidat**.

Načítání informací z monitorovaných licenčního serverů zajišťuje skript `licenses.php` umístěný v adresáři `controller`. Zadejte tedy jeho název za URL adresu úvodní strany a potvrďte jeho vykonání. Rychlost prvního vykonání je závislá na množství démonů dodavatelů softwaru potažmo počtu spravovaných features. V případě, že nebude zobrazeno žádné chybové hlášení, znamená to, že všechny operace byly úspěšně dokončeny. V takovém případě můžete přistoupit k nastavení naplánované úlohy, která zajistí pravidelné vykonávání tohoto skriptu. Naplánované úlohy se v UNIXových operačních systémech realizují prostřednictvím programu `cron`. Obsah konfiguračního souboru pro `cron` zobrazíte příkazem `crontab -e`. Přidáním následujícího řádku:

```
*/5 * * * * wget http://<doména>/licenses >/dev/null 2>&1
```

zajistíte vykonávání skriptu `licenses.php` každých pět minut s potlačením chybových hlášení. V operačním systému Windows se pro účely pravidelného spouštění PHP skriptů používá Plánovač úloh spolu s rozšířením Wget<sup>16</sup>.

Závěrem instalace přidejte prostřednictvím administrační sekce uživatele s právy administrátora a po obdržení registračního e-mailu mu nastavte novou heslovou frázi. Vyzkoušejte přihlášení pomocí uživatelského jména a nově zadaného hesla. Nyní nezapomeňte nastavit konfigurační direktivu `__DEBUG` na `false`! Instalace a konfigurace této webové aplikace je tímto krokem ukončena.

## 5.4 Analýza požadavků na velikost diskového prostoru

Velikost samotné webové aplikace nepředstavuje se svými cca 7 MB zvýšené nároky na velikost diskového prostoru. Jak se později ukázalo, mnohem důležitější bylo optimalizovat velikost databáze zejména s ohledem na historii využití. Za tímto účelem byl upraven mechanismus archivace dat, kdy jsou ukládány pouze skutečně využívané licence. Předtím docházelo k plýtvání volným místem, kdy byly zaznamenávány i licence s nulovým využitím. Po aktualizaci zmíněného mechanismu došlo k téměř 97% snížení nároků na volné místo. Následující analýza předpokládané velikosti databáze po jednom roce provozu této webové aplikace závisí na budoucím

<sup>16</sup> <http://gnuwin32.sourceforge.net/packages/wget.htm>

využití licenčního serveru a vychází z jeho dvoutýdenního sledování. Predikovaná velikost databáze bude pravděpodobně ještě nižší vzhledem k faktu, že během letních prázdnin a zkouškových období neprobíhá standardní výuka.

Následující výpočty platí pro 200 monitorovaných features s periodou sledování 5 minut po dobu 1 kalendářního roku.

- Původně: 200 řádků (v tabulce `licenses_usage`) každých 5 minut + 200 řádků (v tabulce `licenses_available`) denně. Celkové roční využití databáze by se pohybovalo okolo **917 MB**!
- Nyní: v průměru 6 řádků (v tabulce `licenses_usage`) každých 5 minut + 200 řádků (v tabulce `licenses_available`) denně. V takovém případě předpokládám roční využití databáze ve výši cca **30 MB**, které vychází z průměrného týdenního využití 500 kB.

Součástí analýzy náročnosti navržené webové aplikace na velikost diskového prostoru serveru musí být také velikost cache pro grafy. Na velikosti grafického souboru PNG se podílí počet vykreslovaných (zaznamenaných) hodnot využití. Následující kalkulace platí pro implicitní rozměry ( $800 \times 400$  px) generovaných grafů.

- Průměrná velikost grafického souboru se pohybuje okolo **12,4 kB**.
- Maximální počet uložených souborů pro 1 feature nebo 1 démona odpovídá počtu období (tj. 4) zvětšený o 1 grafický soubor se všemi dostupnými daty.
- Velikost cache adresáře se bude postupně zvětšovat podle požadavků na vykreslení historie využití jednotlivých features a démonů pro 5 různých časových období.
- Průměrnou velikost adresáře lze pro 200 features a 8 démonů spočítat jako:  

$$((\text{počet features} + \text{počet démonů}) \times \text{počet období}) \times \text{průměrná velikost souboru}$$

$$\Rightarrow ((200 + 8) \times 5) \times 12,4 = \mathbf{12\,896\,kB} \doteq \mathbf{12,6\,MB}.$$
- Dá se očekávat, že průměrná velikost souboru se bude s přibývajícími daty o využití licenčního serveru zvětšovat. Nepředpokládám však, že by se tento nárůst projevil na velikosti cache adresáře více než trojnásobně.

## 5.5 Testování aplikace v průběhu vývoje

Vývoj webové aplikace probíhal na vlastním počítači s nainstalovaným webovým serverem Apache spolu s databází MySQL a interpretem jazyka PHP.

Načítání zkušební dat o provozu licenčního serveru probíhalo parsováním periodicky generovaného reportu, protože přímé spojení s licenčním serverem nebylo možné.

Testování jednotlivých modulů bylo prováděno paralelně s vývojem. Převážnou většinu chyb se tak podařilo odladit ještě před nasazením na produkční server. Díky použití návrhového vzoru MVC stačilo nalezené chyby opravit pouze jednou, přičemž ostatní moduly nebyly těmito zásahy negativně ovlivněny. Postupným rozšiřováním webové aplikace sice nevznikaly problémy s nekompatibilitou jednotlivých modulů, bylo ji však nutné optimalizovat pro různé webové prohlížeče. Zejména potom zajistit přístupnost i pro uživatele se zakázaným JavaScriptem.

Ještě před implementací aplikace na licenční server jsem otestoval její chování v případě zvýšené zátěže. Pro simulaci zátěže jsem použil nástroj Apache Benchmarking (ab), který je součástí instalace webového serveru Apache. Testování probíhalo nastavením 100 přístupů na konkrétní webové stránky, přičemž bylo povoleno 10 paralelních přístupů. Uvedené parametry testování nastavuje příkaz: `ab -n 100 -c 10 <adresa_testované_stránky>`. Výsledné hodnoty zpoždění obsahuje Tabulka 1. Pro účely testování byly vybrány nejvíce vytěžované moduly a spolu s nimi určeny déletrvající metody.

*Tabulka 1: Doba zpracování jednotlivých požadavků naměřená programem ab*

Adresa	Minimum [ms]	Průměr [ms]	Odchylka $\sigma^{17}$ [ms]	Maximum [ms]
index	489	646	36,6	773
realtime/display	138	241	53,6	404
monitor/display	667	1 131	70,0	1 258
expire/display	115	235	61,9	386

Výsledky syntetických testů prokázaly, že navržená webová aplikace je odolná vůči zvýšené zátěži. Velikost směrodatných odchylek vypovídá o její velmi dobré stabilitě. Mírně vybočuje řadič `monitor.php` při požadavku na zobrazení seznamu features seřazených podle svých démonů v rámci historie využití. Doba zpracování se v takovém případě prodlužuje s narůstajícím počtem paralelních požadavků. Uvedené zpoždění je pravděpodobně způsobeno zvýšeným množstvím dotazů na databázi, která nepodporuje transakční zpracování.

<sup>17</sup> Symbolizuje výběrovou směrodatnou odchylku, která vyjadřuje rozptyl hodnot kolem aritmetického průměru.



## 5.6 Měření výkonu na licenčním serveru

Vyšší vypovídací hodnotu o výkonu webové aplikace poskytuje až měření na produkčním serveru. Proto byla za tímto účelem provedena série testů, jejichž výstupem je nejvyšší množství alokované paměti v průběhu vykonávání jednotlivých požadavků a čas potřebný pro jejich zpracování. Měření rychlosti vykonávání jednotlivých požadavků zajišťuje PHP funkce `microtime()`. Maximální množství alokované paměti při vykonávání skriptů vrací funkce

`memory_get_peak_usage()`. Výsledky testování jednotlivých modulů shrnuje Tabulka 2.

Testování probíhalo za běžného provozu licenčního serveru v průběhu 1 týdne. Testovací konfigurace licenčního serveru je následující:

- procesor: AMD Phenom™ II X3 705e disponující 3 jádry s frekvencí 2,5 GHz
- operační paměť: 8 GB DDR2-800 s podporou technologie ECC
- datové úložiště: 2 × 500 GB WD RE SATA zapojené do svazku RAID 1
- konektivita: 1 Gb/s do sítě LIANE
- operační systém: openSUSE verze 11.4 (x86\_64) s kódovým označením Celadon
- webový server: Apache verze 2.2.17.

Z výsledků měření rychlosti zpracování požadavků na licenčním serveru vyplývá, že doba zpracování je z velké části ovlivněna výkonem HW. Oproti syntetickému měření programem `ab` je webová aplikace několikanásobně rychlejší, což vypovídá mj. o nízkém vytížení licenčního serveru.

Nejvyšší výběrová směrodatná odchylka byla naměřena při vykonávání skriptu `licenses.php`, který zajišťuje pravidelnou analýzu a archivaci využití licenčního serveru. Zvýšená hodnota je způsobena rozdílným využitím serveru v okamžicích zpracování skriptu zejména s ohledem na vytížení databáze. Pochopitelně nejvyšší množství paměti alokuje modul `graphs.php` při požadavku na vykreslení historie využití konkrétní feature nebo démona.

Tabulka 2: Výsledky měření rychlosti webové aplikace na licenčním serveru

Adresa	Doba zpracování [ms]				Nejvyšší využití RAM [kB]
	Minimum	Průměr	Odchylka $\sigma^{17}$	Maximum	
index	117,5	149,4	11,3	240,4	10 430,0
realtime/display	14,2	40,9	14,1	160,2	1 790,0
monitor/display	51,2	122,3	58,7	347,5	1 592,9
expire/display	33,7	42,7	20,2	140,2	1 771,5
graphs/image_*	117,5	159,2	29,6	260,0	11 608,1
licenses	475,9	569,5	84,5	861,7	1 662,2

## Závěr

Vzhledem k potřebám Fakulty mechatroniky, informatiky a mezioborových studií sledovat historii využití licenčního serveru byla v rámci této bakalářské práce navržena příslušná webová aplikace. Aplikace spolupracuje s licenčním serverem FLEXIm a poskytuje registrovaným uživatelům informace o historii využití formou grafů pro jednotlivá pevně stanovená časová období.

Nedílnou součástí aplikace se stal modul, který poskytuje informace o aktuálně zapůjčených licencích včetně informací o vytížení jednotlivých démonů. Tato funkce poskytuje administrátorovi licenčního serveru okamžitý přehled o dění na licenčním serveru. Samozřejmostí byla realizace funkce, která umožňuje zobrazit seznam zakoupených licencí včetně příslušných statistických informací.

Prostřednictvím administrační sekce je umožněno správci licenčního serveru přidávat uživatele, spravovat nastavení licenčních serverů nebo sledovat velikost databáze. V případě problémů s distribucí plovoucích síťových licencí má administrátor licenčního serveru k dispozici nástroj, který mu pomůže zjistit příčinu problému.

Přehledová strana by se dala rozšířit o kanál, který by informoval registrované uživatele o zamítnutých žádostech o zapůjčení licencí. Webová aplikace by mohla eventuálně nabídnout uživateli možnost sledovat vytížení konkrétního démona dodavatele nebo feature rovnou na úvodní straně. V souvislosti s historií využití by její uživatelé mohli do budoucna jistě zajímat i využití v nastavitelném časovém rozmezí. Graficky zobrazovaná data by se dále mohla na požádání vypisovat do tabulky nebo rovnou exportovat např. do široce podporovaného formátu CSV.

S ohledem na licence, kterým brzy skončí platnost, by bylo vhodné v budoucnu realizovat funkci, která by na tuto skutečnost automaticky upozornila správce licenčního serveru např. odesláním informačního e-mailu. Webová aplikace by se z pohledu administrátora dala rozšířit např. implementací modulu, který by umožňoval vzdáleně zastavovat, spouštět nebo restartovat jednotlivé demony dodavatele softwaru.

Jejím nasazením na licenční server FM TUL jsem ověřil funkčnost navrženého řešení. Díky průběžnému sledování využití licenčního serveru existuje nyní možnost, jak optimalizovat využití jednotlivých licencí a předcházet tak jejich nedostatku. Zároveň lze pomocí navržené aplikace vyhodnotit množství skutečně používaných licencí. Předpokládáme, že příštím nákupem nových licencí tak univerzita ušetří nemalé finanční prostředky.

## Seznam použité literatury

- [1] JÁNĚ, Lukáš. *Plánování licencí v systému FLEXlm* [online]. 2011 [cit. 2012-01-13]. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Luděk Matyska. Dostupné z: [http://is.muni.cz/th/325264/fi\\_b/](http://is.muni.cz/th/325264/fi_b/)
- [2] Příručka licence. In: *Příručka licence - Autodesk Exchange* [online]. San Rafael, California (USA), © 2011 [cit. 2012-03-16]. Dostupné z: [http://exchange.autodesk.com/autocad/sites/default/files/autocad\\_pdf\\_lic\\_guide\\_csy\\_0.pdf](http://exchange.autodesk.com/autocad/sites/default/files/autocad_pdf_lic_guide_csy_0.pdf)
- [3] Product Support: FLEXnet Guide. *MathWorks: MATLAB and Simulink for Technical Computing* [online]. Natick, Massachusetts (USA), © 1994-2012 [cit. 2012-03-05]. Dostupné z: <http://www.mathworks.com/support/tech-notes/1300/1303.html>
- [4] KEARSEY, Steve. FLEXlm Seminar. *University of Cambridge Computing Service* [online]. ©2012 [cit. 2012-03-04]. Dostupné z: <http://www-tus.csx.cam.ac.uk/techlink/workshops/flexlm/flexlm.html>
- [5] FlexNet Publisher. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-03-25]. Dostupné z: [http://en.wikipedia.org/wiki/FlexNet\\_Publisher](http://en.wikipedia.org/wiki/FlexNet_Publisher)
- [6] SEJK, Vratislav. Optimalizace licencí podnikového softwaru. *IT Systems* [online]. Brno: CCB s.r.o, 2010, č. 5 [cit. 2012-03-22]. ISSN 1802-002X. Dostupné z: <http://www.systemonline.cz/clanky/optimalizace-licenci-podnikoveho-software.htm>
- [7] PROCHÁZKA, David. *PHP 6: začínáme programovat*. 1. vyd. Pavel Němeček. Praha: Grada, 2012, 183 s. Průvodce (Grada). ISBN 978-80-247-3899-4.
- [8] FlexNet Manager for Engineering Applications. *Flexera Software: Software Licensing, Installation and Application Packaging* [online]. © 2012 [cit. 2012-04-02]. Dostupné z: <http://www.flexerasoftware.com/products/flexnet-manager.htm>

- [9] Enterprise license optimization: šetřete na správném místě. *Unicorn Systems* [online]. © 2012 [cit. 2012-04-02]. Dostupné z: [http://www.unicornsistemas.eu/cz/typova\\_reseni/datasheet-license-optimization.html](http://www.unicornsistemas.eu/cz/typova_reseni/datasheet-license-optimization.html)
- [10] OETIKER, Tobias. About RRDtool. OETIKER+PARTNER AG. *RRDtool* [online]. Olten (Switzerland), 01/17/11 [cit. 2012-04-09]. Dostupné z: <http://oss.oetiker.ch/rrdtool/>
- [11] PHP frameworky. ŠKRÁŠEK, Jan. *Programujte.com* [online]. 21. 2. 2008 [cit. 2012-04-09]. Dostupné z: <http://programujte.com/clanek/2008022000-php-frameworky/>
- [12] VRÁNA, Jakub. *1001 tipů a triků pro PHP*. Vyd. 1. Martin Herodek. Brno: Computer Press, 2010, 456 s. ISBN 978-80-251-2940-1.
- [13] *JpGraph: Most powerful PHP-driven charts* [online]. 2010-10-07 [cit. 2012-03-27]. Dostupné z: <http://jpgraph.net/>

## Seznam příloh

Příloha A – Metoda validující e-mailovou adresu .....	45
Příloha B – Šifrovací funkce.....	46
Příloha C – Úvodní přehledová strana .....	47
Příloha D – Statistika aktuálního využití .....	48
Příloha E – Zobrazení historie využití konkrétní licence .....	49
Příloha F – Platnost zakoupených licencí .....	50
Příloha G – Administrační rozhraní .....	51
Příloha H – ER diagram databáze .....	52

## Příloha A – Metoda validující e-mailovou adresu

```
/**
 * Kontrola platnosti e-mailové adresy
 *
 * @access public
 * @param string e-mailová adresa
 * @return bool status
 */
public function validateEmail($email) {
    $isValid = true;
    $atIndex = strrpos($email, '@');

    if (is_bool($atIndex) && !$atIndex) {
        // neobsahuje zavináč
        $isValid = false;
    } else {
        $domain = substr($email, $atIndex + 1);
        $local = substr($email, 0, $atIndex);
        $localLen = strlen($local);
        $domainLen = strlen($domain);
        if ($localLen < 1 || $localLen > 64) {
            // nepovolená délka místní části
            $isValid = false;
        } else if ($domainLen < 1 || $domainLen > 255) {
            // nepovolená délka doménové části
            $isValid = false;
        } else if ($local[0] == '.' || $local[$localLen - 1] == '.') {
            // místní část začíná nebo končí '.'
            $isValid = false;
        } else if (preg_match('/\\.\.\./', $local)) {
            // místní část obsahuje nejméně 2 po sobě jdoucí '.'
            $isValid = false;
        } else if (!preg_match('/^[A-Za-z0-9\\-\\.]+$/', $domain)) {
            // nepovolený znak v doménové části
            $isValid = false;
        } else if (preg_match('/\\.\.\./', $domain)) {
            // doménová část obsahuje nejméně 2 po sobě jdoucí '.'
            $isValid = false;
        } else if (!preg_match('/^(\\\\.|[A-Za-z0-9!#%&`_=\|/$\'*+?^{}|~.-])+$/ ',
            str_replace("\\\\", '\\', $local)))
        {
            // neplatný znak v místní části, pokud je uvedena
            if (!preg_match('/^"(\\\\.|[^\"])+"$' ', str_replace("\\\\", '\\', $local)))
            { $isValid = false; }
        }
    }

    if ($isValid && !(checkdnsrr($domain, 'MX') || checkdnsrr($domain, 'A')))
    {
        // doména nemá platný DNS záznam
        $isValid = false;
    }
}

return $isValid;
}
```

## Příloha B – Šifrovací funkce

```
/**
 * Šifrování zvolené heslové fráze
 *
 * @access public
 * @param heslová fráze
 * @return otisk heslové fráze
 */
public function hash_password($password) {
    // získání 256 náhodných bitů
    $salt = bin2hex(mcrypt_create_iv(32, MCRYPT_DEV_URANDOM));
    // otisk heslové fráze v kombinaci se získanou 'solí'
    $password_hash = hash('sha256', $salt . $password);
    // uchování 'solí' společně s otiskem heslové fráze
    $hash = $salt . $password_hash;

    return $hash;
}
```



Monitoring licenčního serveru

David Tauchmann  
změnit heslo

Odhlásit

Přehled

Aktuální využití

Historie využití

Platnost licencí

Administrace

Přihlašování dokončeno.

Informace o serveru

Označení

Port@hostname

Stav

Verze lngcmd

shaheen

1717@shaheen.ite.tul.cz

OK

11.10

Využití features pro tento den

Feature	Nejvyšší využití
MATLAB	82 %
Instr_Control_Toolbox	40 %
Signal_Toolbox	40 %
Control_Toolbox	32 %
SIMULINK	22 %
System_Edition	10 %
ISE	10 %
PlanAhead	10 %
ChipscopePro	5 %
ISIM	5 %
com.sigasi.hdt	5 %
Fuzzy_Toolbox	4 %

Statistika feature s nejvyšším využitím pro tento den

MATLAB

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky  
a mezipředmětových studií

Naposledy aktualizováno 24. 04. 2012, 23:45

O aplikaci

*Obr. C: Snímek úvodní strany webové aplikace*

## Příloha D – Statistika aktuálního využití

Monitoring licenčního serveru

David Tauchmann

změnit heslo

Odhlásit

Přehled

Aktuální využití

Historie využití

Platnost licencí

Administrace

Využití démonů na serveru shaheen

Celkové využití licenčního serveru je **14,13 %**.

Démon	Port@server	Stav	Nejvyšší využití	Celkové využití	Verze démona
MLM	27000@shaheen.ite.tul.cz	OK	26 %	7,78 %	11.9
sigasi	50053@shaheen.ite.tul.cz	OK	5 %	5 %	11.10
xilinxd	50050@shaheen.ite.tul.cz	OK	5 %	1,1 %	11.6
tidspssw	50052@shaheen.ite.tul.cz	OK	1 %	0,25 %	11.4
actimgrd	50054@shaheen.ite.tul.cz	OK	0 %	0 %	11.4
adskflex	2080@shaheen.ite.tul.cz	OK	0 %	0 %	11.9
mgcld	50051@shaheen.ite.tul.cz	OK	0 %	0 %	11.6
snpslmd	50055@shaheen.ite.tul.cz	OK	0 %	0 %	11.6

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky

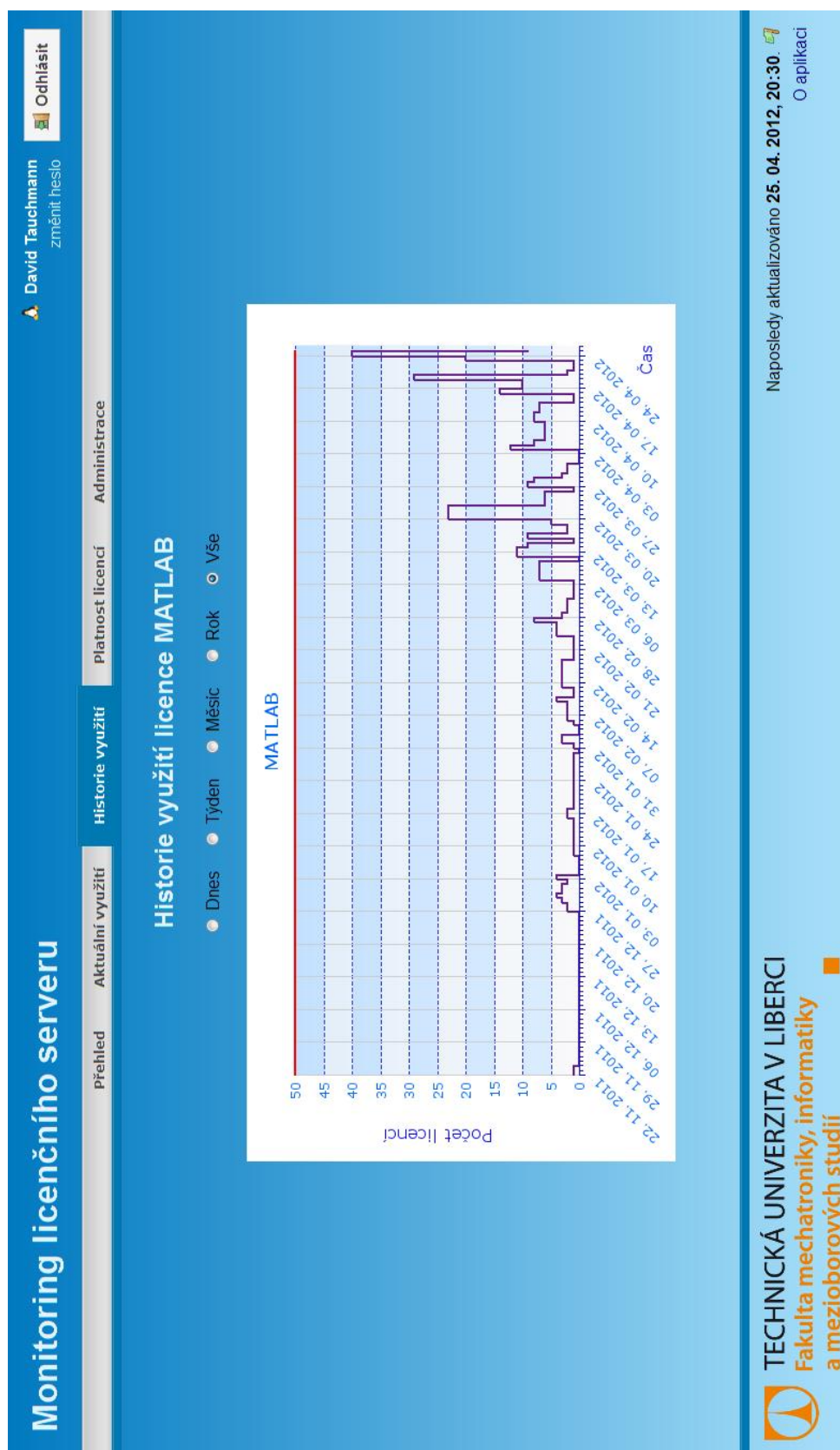
a meziporobových studií

Naposledy aktualizováno 02. 05. 2012, 11:35

O aplikaci

Obr. D: Snímek informující o využití licenčního serveru

## Příloha E – Zobrazení historie využití konkrétní licence



Obr. E: Snímek obrazovky s grafem historie využití

## Příloha F – Platnost zakoupených licencí

Monitoring licenčního serveru

David Tauchmann  
změnit heslo

Odhlásit

Přehled

Aktuální využití

Historie využití

Platnost licencí

Administrace

Platnost licencí na serveru shaheen

Celkový počet features s omezenou platností je 58.

Démon	Port@hostname	Počet features	S omezenou platností	Nejbližší expirace	Po splatnosti
sigasi	50053@shaheen.ite.tul.cz	1	1	15. 09. 2012	0
mgcid	50051@shaheen.ite.tul.cz	50	50	08. 03. 2013	0
snpslmd	50055@shaheen.ite.tul.cz	5	5	08. 03. 2013	0
actlmgrd	50054@shaheen.ite.tul.cz	52	2	08. 03. 2013	0
xilinxd	50050@shaheen.ite.tul.cz	44	0	nikdy	—
tidspssw	50052@shaheen.ite.tul.cz	4	0	nikdy	—
MLM	27000@shaheen.ite.tul.cz	15	0	nikdy	—
adskflex	2080@shaheen.ite.tul.cz	22	0	nikdy	—

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky  
a mezipřoborových studií

Naposledy aktualizováno 25. 04. 2012, 21:10.

O aplikaci

Obr. F: Snímek zachycující platnost licencí

## Příloha G – Administrační rozhraní

Monitoring licenčního serveru

Přehled

Aktuální využití

Historie využití

Platnost licencí

Administrace

David Tauchmann

změnit heslo

Odhlásit

Administrace serveru

Označení	Port@hostname	Stav	Verze	Imgrd
<input type="radio"/> shaheen	1717@shaheen.ite.tul.cz	OK		11.10

Přidat

Změnit


Odstranit

Přidat uživatele

Registrace

Velikost databáze

321,06 kB

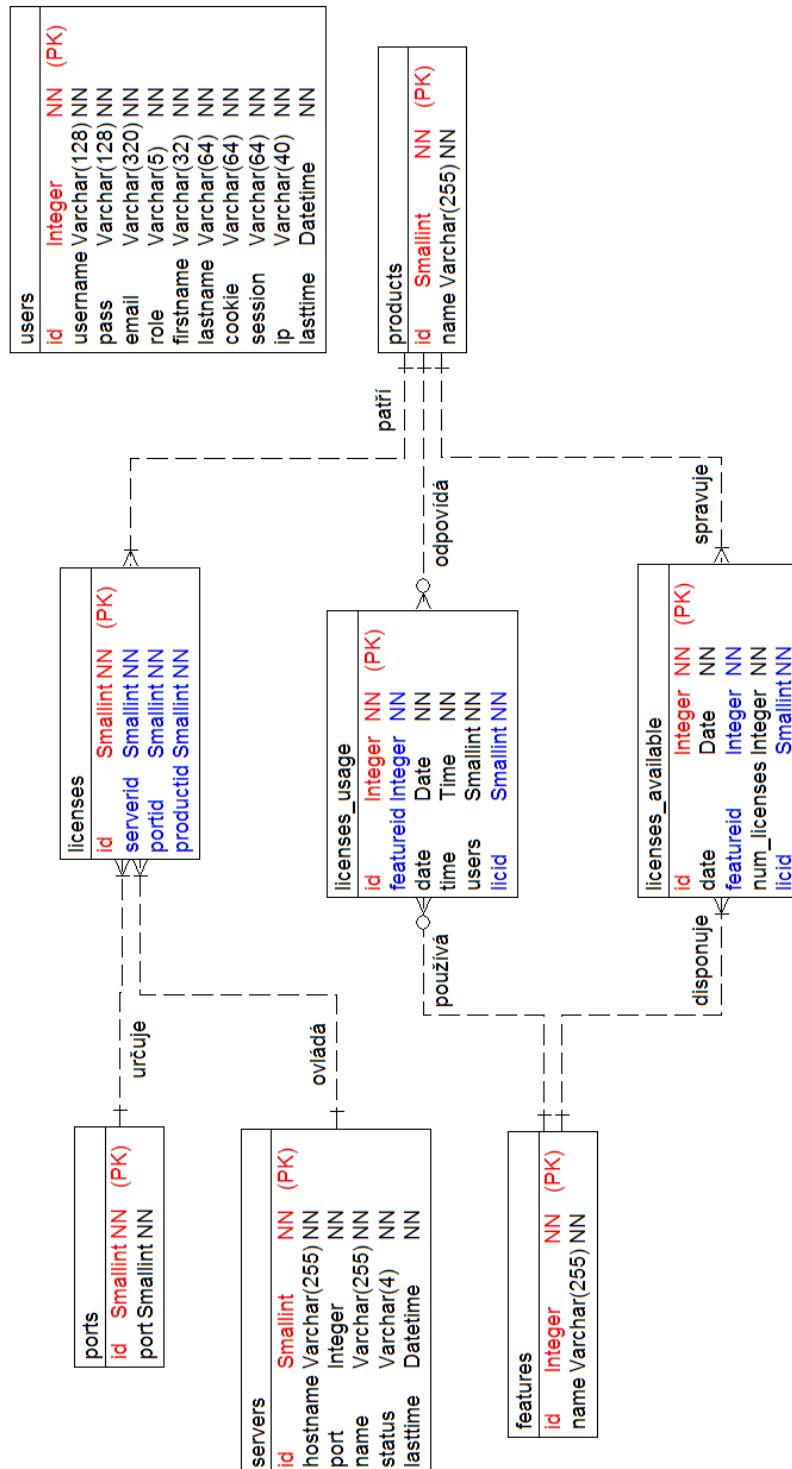
 TECHNICKÁ UNIVERZITA V LIBERCI  
Fakulta mechatroniky, informatiky  
a mezioborových studií

Naposledy aktualizováno 25. 04. 2012, 21:30

O aplikaci

Obr. G: Snímek popisující rozložení administračního rozhraní

## Příloha H – ER diagram databáze



Obr. H: Databázový diagram popisující vztahy mezi entitami